BITS - Pilani,Hyderabad Campus

CS F469 IR Assignment - 1

Deadline: 9/9/2018

This assignment is aimed at designing and developing one's own text based information retrieval system. The assignment aims at building the searching module according to Boolean, Vector Space or Probabilistic model of Information Retrieval.

The assignment can be done in groups of **at most 4 (Four) members**. All the group members are expected to contribute to all the aspects of the assignment namely, design, implementation, documentation and testing.

Programming Languages:

The assignment can be implemented in any programming language of your choice. STL's and inbuilt packages can be used only for Normalization (C++'s Boost Library, Python's NLTK Package etc.). You are expected to code the core functionality of the model that you choose (TF-IDF in case of Vector Space model etc.)

The following are three tasks descriptions:

(Implement any 1)

1. Plagiarism Checker

   The task is to build a plagiarism checker which will rank documents based on similarity. The program should build its indexes and IR model based on a set of training corpus and do all the preprocessing which it deems necessary. Then the model should take another document from the test set and should do either of the two things:

   - Compute how much percentage of the document is unique and should list documents from where the content has been copied.
   - Rank all the training set documents with respect to closeness from the test document in consideration.

Kindly note that similarity score formula while considering query as a document while change slightly. One cannot ignore the tf of the query in this case. Detailed explanation of the results should be explained in the **design document.** You should mention briefly about cases which would be successfully recognized as well as cases which won't be.

2. Domain Specific Information Retrieval System

The task is to build a search engine which will cater to the needs of a particular domain. You have to feed your IR model with documents containing information about the chosen domain. It will then process the data and build indexes. Once this is done, the user will give a query as an input. You are supposed to return top 10 relevant documents as the output. Your results should be explainable. The design document should clearly explain the working of the model along with detailed explanation of any formulas that you might have used. For Eg:

- You can build a search engine for searching song lyrics. Dataset: https://labrosa.ee.columbia.edu/millionsong/musixmatch
- Other domains on which you can find datasets easily are:
    - Finance
    - Health
    - Automobiles etc.

You can also scrape data from the web using the scraper described below in the additional resources. You are free to use your own web scraper as well.

3. Profession Specific Information Retrieval System

The task is to build a search engine which will cater to the needs of a particular profession. It will then process the data and build indexes. Once this is done, the user will give a query as an input. You are supposed to return top 10 relevant documents as the output. Your results should be explainable. The design document should clearly explain the working of the model along with detailed explanation of any formulas that you might have used. For Eg:

- Westlaw is a search engine which caters to the need of lawyers. You can also scrape data from the web using the scraper described below in the additional resources. You are free to use your own web scraper as well.

Additional Resources:

1. Stemming:
    a. Martin Porter's 'Porter Stemmer' can be used for this purpose. Implementation in multiple languages can be found in the above link.
2. Tokenization:
    a. For this step you can use any standard tokenizer or inbuilt package. Following are a few sources:
        i. Python's NLTK package.
        ii. Stanford Tokenizer.
        iii. TM package of R.
3. Datasets:
    - Domain specific datasets can be found at: https://snap.stanford.edu/data/index.html
    - Corpus for Plagiarism Checker: http://ir.shef.ac.uk/cloughie/resources/plagiarism_corpus.html
    - https://www.springboard.com/blog/free-public-data-sets-data-science-project/
    - https://www.dataquest.io/blog/free-datasets-for-projects/
    - Teams are allowed to use their own corpus

4. Details about the scrapper: Refer to Lecture 10 of Impartus

5. Incase of any queries kindly mail to Ms.Shreya Reddy Nimma (f20150951@hyderabad.bits-pilani.ac.in)

6. Incase you wish to build a Plagiarism Checker the documents will be provided in 2-3 days time on CMS

Deliverables:

The final submission must contain the following documents:

1. **Design Document** – This document should contain the description of the application's architecture along with the major data structures used in the project. Precision and Recall, if possible, should also be calculated. Running for all the preprocessing should be mentioned. Also mention the running time for search or retrieval.
2. **Code** – The code should be well commented.
3. **Documentation** – All the classes, functions and modules of the code must be documented. Software that automatically generate such documents can be used – pydoc for Python, Eclipse for Java etc.

4. **README** – The README file should describe the procedure to compile and run your code for various datasets.

Submission Guidelines:

All the deliverables must be zipped and submitted to bphc.information.retrieval@gmail.com latest by **deadline.**

You are expected to demo your application and present your results as per the schedule that will be made available.

Evaluation Criteria for Task :

S.No.	Task	Marks
1.	Tokenization and Normalization	5
2.	Efficient usage of Data Structures with justification	10
3.	Index Construction	5
4.	Accurate Data Retrieval	5
5.	Viva	5
6.	Novelty / Out-of-the-box thinking (Anything that is not covered in the lectures.)	10
	Total	**40**

It should be noted that all the assignments would be run through a plagiarism detector and based on the results, the marks would be altered. The final decision lies in the hand of the instructor and only one submission per group would be allowed for one assignment.